

Información Contextual en Ambientes MCH

Sergio Martig Perla Señas

Laboratorio De Investigación y Desarrollo en Informática y Educación (LIDInE)

Instituto de Investigación en Ciencias y Tecnología Informática (IICyTI)

Dpto. de Ciencias de la Computación

Universidad Nacional del Sur – Bahía Blanca

{smartig,psenas}@cs.uns.edu.ar}

Resumen

En un Mapa Conceptual Hipermedial (MCH) se plasman los conceptos fundamentales de un tema y sus relaciones, particionándolos y jerarquizándolos en vistas. El saldo positivo que se logra con dicha partición puede tornarse negativo si el lector del mapa pierde las referencias conceptuales con respecto al todo, mientras centra su atención en una vista particular. Esta es la principal motivación para ampliar el ambiente MCH existente con un Grafo de Vistas (GV_{MCH}) que brinde al usuario la información contextual mientras focaliza su atención en una vista particular. Se presenta aquí la definición y los algoritmos para la generación del GV_{MCH} .

Palabras claves

Entornos de Aprendizaje Significativo - Mapa Conceptual Hipermedial - Grafo de Vistas – Contexto

1 Introducción

Los Mapas Conceptuales Hipermediales (MCH) son esquemas de representación de conocimiento, suman la riqueza de los Mapas Conceptuales de Novak (MC) con el potencial de la tecnología hipermedial. Junto con la Plataforma que hace flexible su diseño y mantenimiento, constituyen una poderosa herramienta para la construcción de conocimiento. En el área educativa, han sido probados con éxito como potentes estructuras capaces de contribuir con la construcción de aprendizajes significativos [9]. Un ambiente de aprendizaje MCH permite trabajar expresamente sobre la relación correcta entre el conocimiento recientemente aprendido y los conocimientos previos o entre diversos enfoques de un mismo tema. Se trata de una propuesta que trasciende la mera simplificación de las tareas en los procesos de enseñanza-aprendizaje, y se presenta como una herramienta efectiva para la construcción del conocimiento, de acuerdo con un enfoque constructivista.

En un MCH se plasman los conceptos fundamentales de un tema y sus relaciones. Se destaca especialmente el trabajo de selección y jerarquización de los conceptos para la organización de las vistas. Esto y la posibilidad de establecer relaciones cruzadas diferencian a los MCH de otras propuestas que asocian MC con hipermedia. El saldo positivo que se logra con la partición del mapa en vistas (desde lo pedagógico y desde lo operacional), puede tornarse negativo si el lector del mapa pierde las referencias conceptuales con respecto al todo mientras centra su atención en una vista particular. Esta es la principal motivación para ampliar el ambiente MCH existente con un Grafo de Vistas (GV_{MCH}) que brinde al usuario la información contextual mientras focaliza su atención en una vista particular. Se presentan en este trabajo la definición y los algoritmos para la generación del GV_{MCH} , distinguiéndose dos situaciones: la construcción a partir de un MCH ya creado y su generación en forma automática durante el proceso de autoría.

2 Mapas Conceptuales Hipermediales

Los MC son una herramienta cuya potencialidad ha sido probada en el logro de aprendizajes significativos, para poder asumir la propia elaboración de significados, en definitiva para poder “aprender

a aprender”. En los últimos años se ha incrementado su uso en las aulas, no sólo como herramienta de aprendizaje sino también como técnica de evaluación. Sin embargo, se ha detectado que existen aspectos operacionales que dificultan la realización de estos mapas, si se piensa en su construcción mediante elementos tradicionales, especialmente cuando el mapa cuenta con un elevado número de conceptos, o cuando es necesario hacer refinamientos del mismo o cuando se desea interconectar mapas ya existentes. La aplicación de tecnología informática hipermedial facilita la construcción de los MC, salvando las dificultades antes mencionadas. Además y principalmente, en los MCH, se favorece sustancialmente el trabajo de selección y de jerarquización de conceptos, enriqueciendo de esta manera su valor pedagógico. Siguiendo esta motivación es que se crearon los MCH, una metodología de desarrollo y una plataforma específica que permite la creación, mantenimiento, lectura e interconexión de estos mapas.[5]

Un MCH es un documento hipermedial; cada nodo de la hipermedia contiene una colección de conceptos relacionados entre sí por palabras enlace. A cada uno de estos nodos se lo denomina *vista* del MCH. Cada vista puede ser visualizada en una ventana y es caracterizada por un color y por un nombre, por defecto el del concepto propio más abarcativo en dicha vista. Se considera que una vista debería contar entre siete y diez conceptos como máximo. Desde un punto de vista psicológico, esta limitación tiene relación con el número máximo de bloques de información que se pueden retener después de un breve intervalo de percepción [6]. Desde un punto de vista práctico, se trata de tener una visión panorámica (en una sola pantalla) de los conceptos más relevantes y sus relaciones. Aunque estas reglas suenan arbitrarias desde la perspectiva de la representación del conocimiento, es importante destacar que la partición del mapa en vistas está más relacionada con los temas de visualización y percepción por parte del usuario de un MC que con el del modelamiento de la realidad.

El saldo positivo que se logra con la partición del mapa en vistas (desde lo pedagógico y desde lo operacional), puede tornarse negativo si el lector del mapa pierde las referencias conceptuales con respecto al todo mientras centra su atención en una vista particular

Si bien la metodología propuesta para la creación de los MCH subsana los aspectos operacionales que dificultan la creación de los MC tradicionales, introduce las características propias de la hipermedia.

3 Grafo de las Vistas de un MCH: GV_{MCH}

En un ambiente MCH el usuario (autor/lector) al recorrer el mapa, ya sea durante el proceso de creación o de consulta, no tiene la visión simultánea de todas las vistas del mapa. Es decir el autor/lector tiene foco sobre la vista actual, pudiéndole surgir la necesidad de tener visible el contexto de esa vista dentro del MCH en cuestión.

Una manera de brindar el contexto de una vista es mostrar las relaciones existentes entre todas las vistas que componen el MCH, para que el usuario al consultar ese mapa de las vistas del MCH pueda extraer la información contextual que necesite. Esta es la motivación para la creación del Grafo de Vistas de un MCH (GV_{MCH}), el cual es un grafo con las siguientes características:

- Es un grafo jerárquico y rotulado. La ubicación de los vértices respeta la jerarquía de vistas del MCH original.
- Cada vértice del GV_{MCH} es la representación de una vista del MCH y tiene las siguientes características: se rotula con el nombre y se pinta con el color de la vista que representa
- Un arco entre dos vértices V_{v1} y V_{v2} indica la existencia de al menos una relación entre algún concepto de la vista $V1$ y alguno de la vista $V2$ del MCH.

Se debe destacar que los niveles asociados a los vértices del GV son totalmente independientes de los niveles de los conceptos contenidos en las distintas vistas. La semántica del GV es diferente a la del

GI_{MCH} [4]. Mientras el primero representa el mapa de las vistas de un MCH, el segundo brinda una síntesis del mismo como resultado de la integración de todas las vistas que lo componen.

En la Fig. 1 se muestran algunas vistas de un MCH sobre el Sistema Solar y en la Fig. 2 su GV.

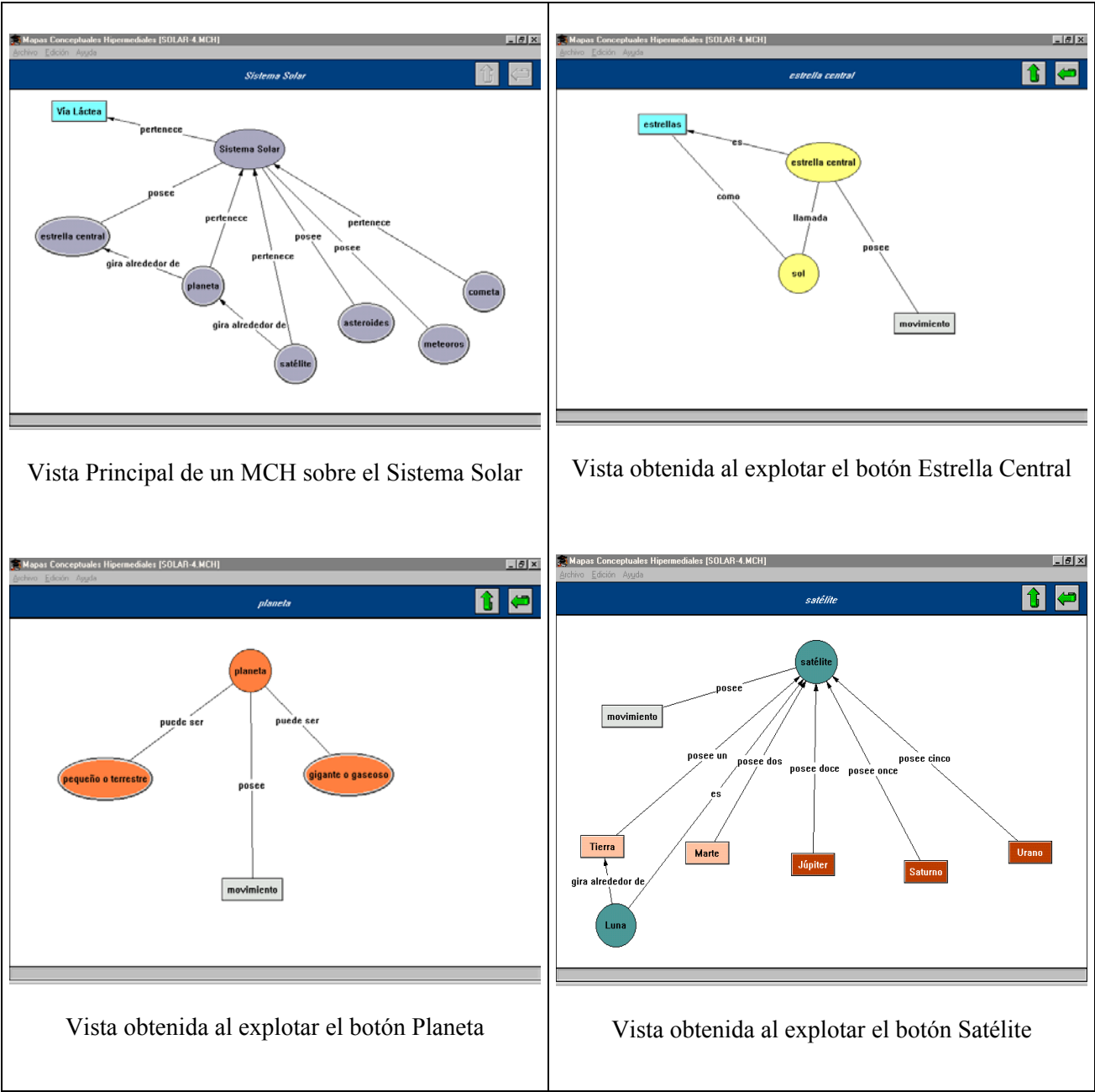


Fig. 1 Cuatro Vistas del MCH sobre el Sistema Solar

3.1 Generación del GV_{MCH}

A continuación se desarrollan los algoritmos para la generación del GV_{MCH}. Se distinguen dos situaciones:

- Su generación a partir de un MCH ya creado.
- Su generación en forma automática durante el proceso de autoría

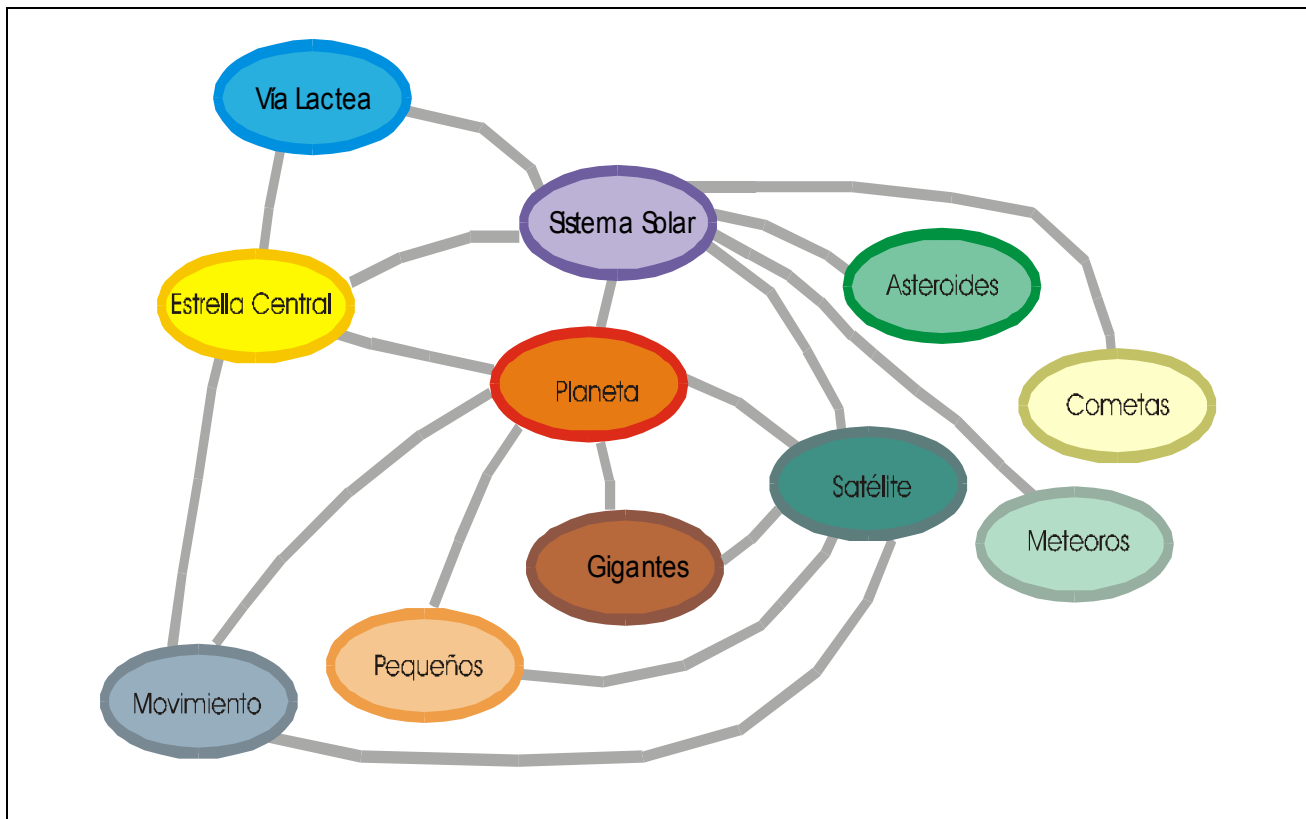


Fig. 2: GV de las vistas presentadas en la figura 1

3.1.1 Generación a partir de un MCH

Para construir el GV_{MCH} debemos establecer cómo determinar la existencia de relaciones entre dos vistas del mismo.

Del análisis de los MCH se pueden distinguir los siguientes elementos:

Elementos		Representación
Vistas		Ventana
Conceptos Propios	Terminales	Elipse Rotulada
	No Terminales	Botón Elíptico Rotulado
Conceptos Importados		Botón Rectangular Rotulado
Relaciones	Internas	Arco Rotulado
	Externas	Arco Rotulado entre rectángulo y elipse
Apariencias	Sonido – Texto	Elipse Marcada con *
	Video –Gráfico	
Ejemplos		Palabras

Dado un MCH M compuesto por las vistas V_1, V_2, \dots, V_n . Una vista V_i está relacionada con otra vista V_j ; para $1 \leq i, j \leq n$, si:

- En la vista V_i aparece un concepto propio C no terminal (botón elíptico) que explota en la vista V_j .
- En la vista V_i aparece un concepto importado C (botón rectangular) que aparece como concepto propio en la vista V_j .

El hecho de que el GV sea un grafo jerárquico implica que hay un nivel asociado con cada vértice V_i que representa a una vista V del MCH. El nivel de un vértice indica la profundidad a la que se encuentra la

vista por él representada dentro del MCH, es decir la cantidad de botones que se deberán explotar durante la navegación del MCH a partir de la vista principal para llegar a la vista V. Los conceptos importados plantean una situación distinta pues puede ocurrir que las vistas en las cuales aparecen como conceptos propios:

- Formen parte del GV por haber sido incorporadas al mismo al explotar algún botón elíptico, en cuyo caso se respeta el nivel asignado oportunamente a la vista que los contienen.
- No pertenezcan al conjunto de vértices del GV, con lo cual se las incorporan con el mismo nivel que el de las vistas que contienen las relaciones externas por las que son alcanzados.

Resumiendo, la finalidad del GV_{MCH} es brindar información contextual a la vista actual (la que está siendo enfocada por el usuario), en tal sentido lo que resulta de interés es mostrar si una vista está antes, en un mismo nivel o después de la vista actual. Otro aspecto a considerar es que dependiendo de la topología del MCH el GV puede no ser único.

Algoritmo de Creación del GV_{MCH} (M: MCH, \underline{GV}_M : Grafo por niveles)

Datos Locales

Nivel: 1..MaxNivel
VistaPrinc: Vista

Acciones:

CrearGrafoNulo(\underline{GV}_M)
Nivel \leftarrow 1
VistaPrincipal(M)
AgregarVérticeVista(VistaPrincipal(M), Nivel, \underline{GV}_M)
IncVistas(VistaPrincipal(M), Nivel, \underline{GV}_M)
IncRelacionesExternas(M, Nivel, \underline{GV}_M)

Algoritmo AgregarVérticeVista (V: Vista, Nivel, \underline{GV}_M : Grafo por niveles)

Datos Locales

NuevoVértice: Vértice

Acciones:

NuevoVértice.Rótulo \leftarrow Nombre(V)
NuevoVértice.Color \leftarrow Color(V)
NuevoVértice.Nivel \leftarrow Nivel
 $\underline{GV}_M.Vértices \leftarrow \underline{GV}_M.Vértices \cup \text{NuevoVértice}$

Algoritmo AgregarArcoVistas (VistaOrigen, VistaDestino: Vista, \underline{GV}_M : Grafo por niveles)

Datos Locales

NArco: TArco

Acciones:

NArco.Origen \leftarrow Vértice(Nombre(VistaOrigen))
NArco.Destino \leftarrow Vértice(Nombre(VistaDestino))
 $\underline{GV}_M.Arcos \leftarrow \underline{GV}_M.Arcos \cup \text{NArco}$

Algoritmo IncVistas(VistaOrigen: Vista, Nivel, GV_M: Grafo por Niveles)

Acciones:

Para cada botón elíptico B de VistaOrigen
IncVérticesyArcos(VistaOrigen,Vista(B),Nivel+1, GV_M)

Algoritmo IncVérticesyArcos(VistaOrigen, VistaDestino: Vista, Nivel, GV_M : Grafo por Niveles)

Acciones:

AgregarVérticeVista(VistaDestino, Nivel, GV_M)
AgregarArcoVistas(VistaOrigen, VistaDestino, GV_M)
IncVistas (VistaDestino, Vértice(V).Nivel, GV_M)

Algoritmo IncRelacionesExternas(M: MCH, Nivel, GV_M: Grafo por Niveles)

Acciones:

Para cada vista V de M
Para cada botón rectangular R de V
Si Vértice(Vista(R)) \notin GV_M.Vértices
Entonces
AgregarVérticeVista (Vista(R), Nivel, GV_M)
IncVistas (Vista(R), Nivel, GV_M)
AgregarArcoVistas(V, Vista(R), GV_M)

3.1.2 Durante la etapa de autoría

Si se opta por generar el GV conjuntamente con el MCH, debemos evaluar cómo inciden en el mismo las distintas operaciones provistas por la plataforma en modo autor.

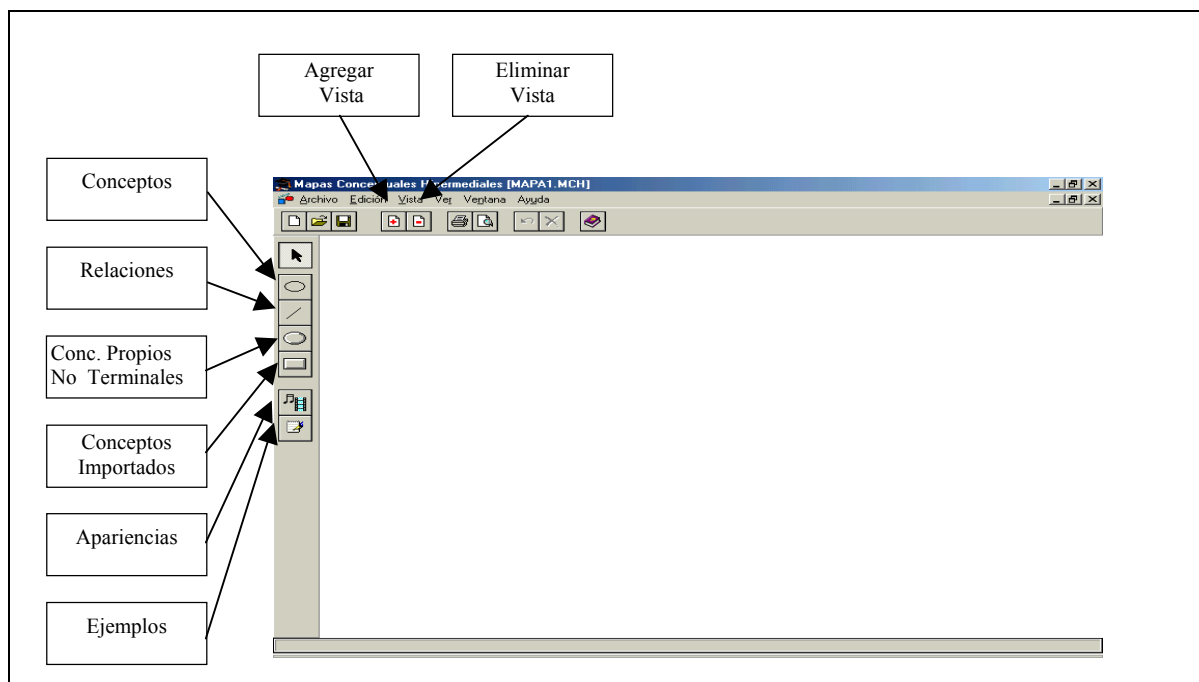


Fig. 3 Pantalla Principal de la Plataforma (modo autor)

De los elementos que uno puede manipular durante la edición de un MCH los que pueden incidir en el GV son: Vistas – Conceptos Propios No Terminales – Conceptos Importados - Relaciones. A continuación se detalla las distintas operaciones que pueden realizarse sobre cada uno de ellos y como repercuten en el GV.

– *Vistas*: La Plataforma ofrece la posibilidad de :

– *Agregar una Vista*: Una vez creada la nueva Vista, simplemente hay que agregar un nuevo vértice al GV.

Algoritmo AgregarVérticeVista

(V: Vista, Nivel, \underline{GV}_M : Grafo por niveles)

Datos Locales

NuevoVértice: Vértice

Acciones:

NuevoVértice.Rótulo \leftarrow Nombre(V)

NuevoVértice.Color \leftarrow Color(V)

NuevoVértice.Nivel \leftarrow Nivel

$\underline{GV}_M.Vértices \leftarrow \underline{GV}_M.Vértices \cup \text{NuevoVértice}$

Para este caso particular el nuevo vértice agregado deberá tener asignado un nivel máximo (ver incorporación de relaciones).

– *Eliminar una Vista*: La Plataforma exige que la vista para ser eliminada este vacía. Por lo tanto las modificaciones al GV se realizan durante la eliminación de los componentes de una vista..

Además de las operaciones mencionadas, a través del menú desplegable Vistas, el autor puede seleccionar:

– *Nombre*: Esta operación le permite editar el Nombre de la vista actual.

– *Color*: Operación que permite cambiar el Color de la vista actual.

Tanto el Nombre como el Color de una vista son atributos del vértice que la representa en el GV. Por lo tanto las operaciones que permiten su modificación durante la edición de una Vista del MCH deben asegurar la actualización de los atributos correspondientes en el vértice que representa a dicha vista.

Algoritmo ModificarVérticeVista(V: Vista, Atributo:(Nombre,Color), \underline{GV}_M : Grafo por niveles)

Acciones:

VerticeVista \leftarrow Vértice(Vista(V))

Si Atributo = Nombre

Entonces

VerticeVista.Rótulo \leftarrow Nombre(V)

Sino

VerticeVista.Color \leftarrow Color(V)

– *Conceptos Propios No Terminales* (Botón Elíptico): La plataforma permite

– *Agregar un botón elíptico*:

El agregado de un botón elíptico B en una vista V_{origen} determina la creación automática de una nueva vista V_{nueva} con un nombre, un color y un nivel que será:

$$\text{Nivel}(V_{nueva}) = \text{Nivel}(V_{origen}) + 1$$

La correspondiente actualización del GV se logra invocando a la operación:

AgregarVérticeVista (V_{nueva} , $\text{Nivel}(V_{origen}) + 1$, $\underline{GV_M}$)

– *Eliminar un botón elíptico:*

La eliminación de un botón elíptico B en una vista V_{origen} que explota en otra vista V_B determina la eliminación de todo el submapa a partir de la vista V_B .

Algoritmo EliminarSubMapa

(V: Vista, $\underline{GV_M}$: Grafo por niveles)

Acciones:

Para cada botón elíptico B de V
 EliminarSubMapa(Vista(B), $\underline{GV_M}$)
Para cada botón rectangular R de V
 EliminarArco(Vista(R), V, $\underline{GV_M}$)
 EliminarVertice(V, $\underline{GV_M}$)

El algoritmo EliminarVértice deberá no solo eliminar el vértice pasado como parámetro sino también todos los arcos que inciden en él:

Algoritmo EliminarVértice(V: Vista, $\underline{GV_M}$: Grafo por niveles)

Acciones:

Para cada arco $A \in \underline{GV_M}.\text{Arcos}$
 Si $A.\text{Origen} = \text{Vértice}(V)$ o
 $A.\text{Destino} = \text{Vértice}(V)$
 Entonces
 Eliminar A de $\underline{GV_M}.\text{Arcos}$
Eliminar Vértice(V) de $\underline{GV_M}.\text{Vértices}$

Algoritmo EliminarArco(V_{origen} , $V_{destino}$: Vista, $\underline{GV_M}$: Grafo por niveles)

Acciones:

Para cada arco $A \in \underline{GV_M}.\text{Arcos}$
 Si $A.\text{Origen} = \text{Vértice}(V_{origen})$ y
 $A.\text{Destino} = \text{Vértice}(V_{destino})$
 Entonces
 Eliminar A de $\underline{GV_M}.\text{Arcos}$

– *Modificar su rótulo:*

La modificación del rótulo de un botón elíptico B determina que cambie el nombre de la vista V_B en la cual explota.

Para mantener la consistencia del GV basta invocar a la operación:

Algoritmo ModificarVérticeVista (V_B , Nombre, $\underline{GV_M}$)

– *Conceptos Importados (Botón Rectangular):*

Durante la etapa de edición de un MCH la incorporación de conceptos importados en una vista no modifica la conectividad del GV_{MCH} pues esto queda supeditado a la inclusión de una relación entre un concepto propio y uno importado.

No sucede lo mismo con la eliminación de un concepto importado, pues esta acción elimina todas las relaciones que involucren a ese concepto. Entonces desde el punto de vista del GV_{MCH} ante una eliminación de un concepto importado en una vista hay que eliminar el arco entre la vistas involucradas:

Algoritmo EliminarArco (V_{Actual} , $Vista(R)$, GV_M)
--

– *Relaciones*

- *Incorporación de Relaciones:* Durante la edición de una vista el autor puede incorporar relaciones internas o externas.

La incorporación de relaciones internas que modifican al GV son las que involucran a un concepto propio no terminal B, determinan la incorporación de un arco entre V y la vista en la cual explota B ($Vista(B)$).

Una relación externa se establece entre un concepto propio y uno importado. Para poder importar en una vista V un concepto R, previamente debió crearse la vista ($Vista(R)$) en la cual R aparece como concepto propio. El hecho de la preexistencia de $Vista(R)$ implica que la misma tenga asignado un nivel, por tal motivo al actualizar el GV no solo hay que incorporar un arco entre las vista V y $Vista(R)$, sino que también hay que verificar el nivel asignado a $Vista(R)$ y eventualmente actualizarlo.

Algoritmo AgregarArcoVistas

(V_{Origen} , $V_{Destino}$: Vista, TipoRelación: (Interna, Externa), GV_M : Grafo por niveles)

Datos Locales

NuevoArco: TArco

Acciones:

NuevoArco.Origén ← Vértice(Nombre(VistaOrigen))

NuevoArco.Destino ← Vértice(Nombre(VistaDestino))

$GV_M.Arcos \leftarrow GV_M.Arcos \cup NuevoArco$

Si TipoRelación = Externa

Entonces

Si Vértice(Nombre(VistaDestino)).Nivel < Vértice(Nombre(VistaOrigen)).Nivel+1

Entonces

Vértice(Nombre(VistaDestino)).Nivel ← Vértice(Nombre(VistaOrigen)).Nivel+1
--

– *Eliminación de Relaciones:*

En forma análoga a la incorporación de relaciones, la eliminación determinará una modificación del GV únicamente cuando uno de los conceptos involucrados en la relación sea un no terminal en la vista. La remoción de una relación R en una vista V, en la cual uno de los conceptos es un botón B, puede originar la eliminación del arco entre los nodos representando a las vistas V y $Vista(B)$. Dado que entre dos vistas puede haber más de una relación la eliminación de una de ellas no siempre origina una modificación del GV.

4 Conclusiones

Al considerarse a la tecnología computacional como una herramienta efectiva en la construcción del conocimiento, han tomado relevancia los Ambientes de Aprendizaje Computacionales. En este marco se inscribe el ambiente MCH que reúne los beneficios de la hipermedia y de propuestas educativas

constructivistas. El enriquecimiento del entorno MCH con un grafo de vistas colabora con los autores y lectores del mapa en el logro de aprendizajes significativos a partir de una mayor comprensión temática.

Referencias

- [1] Ausubel, D. P., Novak J. D. *Educational Psychology: A Cognitive View*. 2nd Ed. New York: Holt, Rinerhart and Winston. 1978.
- [2] Di Battista, P. Eades, G. Tamassia, R. y Tollis, I. “*Graph Drawing: algorithms for the visualization of graphs*”, Prentice Hall, 1999.
- [3] Lajoie, S. “*Computer Environments as Cognitive Tools for Enhancing Learning*”. 1993. McGill University
- [4] Martig, S. y Señas, P. “*Herramientas para la construcción de conocimiento en ambientes de aprendizaje abiertos: Construcción y Visualización del Grafo Integrador de un MCH*”. VI CACIC. Argentina. 2000.
- [5] Moroni, N. - Vitturini, M. - Zanconi, M. - Señas, P. “*Una plataforma para el desarrollo de mapas conceptuales hipermédiales*”. Taller de Software Educativo - IV Jornadas Chilenas de Computación. Valdivia. 1996.
- [6] Novak, J. “*Metalearning and metaknowledge strategies to help students learn how to learn. Cognitive Structure and Conceptual Change*”. New York. Academic Press.1985.
- [7] Señas, P., Moroni, N., Vitturini, M. y Zanconi, M.: “*Hypermedial Conceptual Mapping: A Development Methodology*”. 13th International Conference on Technology and Education. University of Texas at Arlington, Departament of Computer Science an Engineering. New Orleans 1996.
- [8] Señas, P., Moroni, N. “*Computing Environments for metalearning. Interconnecting Hypermedia Concept Maps*”. ED-MEDIA 2000. Montreal. Canada. 2000.
- [9] Zanconi, M., Moroni, N., Vitturini, M., Malet, A., Borel, C. y Señas, P. “*Tecnología computacional y meta-aprendizajes*”. RIBIE-98. 1998.